JVET-O0305
Non-CE4: Undefined condition of merge mode syntax

K. Abe, T. Toma, J. Li, V. Drugeon

Panasonic Corporation

Problem

VTM5.0 spec of merge mode syntax

merge_data( ) {	
regular_merge_flag[x0][y0]	
if (regular_merge_flag[x0][y0] = = 1) {	
merge_idx[x0][y0]	Code regular merge
} else {	
mmvd_merge_flag[x0][y0]	
if (mmvd_merge_flag[x0][y0] = = 1) {	
. . . .	Code MMVD merge
} else {	
merge_subblock_flag[x0][y0]	
if (merge_subblock_flag[x0][y0] = = 1) {	
. . . .	Code sub block merge
} else {	
ciip_flag[x0][y0]	
if (ciip_flag[x0][y0] = = 1) {	
. . . .	Code CIIP merge
} else if (triangle conditions are true){	
. . . .	Code triangle merge
}	
}	
}	
}	
}	

It is not defined when none of the merge modes are selected.

This contribution presents three methods to solve this issue.

Panasonic

Method1

- Coded as regular merge mode when none of the merge modes are selected.

merge_data( ) {	
regular_merge_flag[x0][y0]	
if (!regular_merge_flag[x0][y0]) {	
mmvd_merge_flag[x0][y0]	
if (mmvd_merge_flag[x0][y0] = = 1) {	
. . . .	Code MMVD merge
} else {	
merge_subblock_flag[x0][y0]	
if (merge_subblock_flag[x0][y0] = = 1) {	
. . . .	Code sub block merge
} else {	
ciip_flag[x0][y0]	
if (ciip_flag[x0][y0] = = 1) {	
. . . .	Code CIIP merge
} else if (triangle conditions are true){	
. . . .	Code triangle merge
} else {	
RegularMergeFlag[x0][y0] = 1	Change to regular merge
}	
}	
}	
}	
if (RegularMergeFlag[x0][y0] = = 1) {	
merge_idx[x0][y0]	Code regular merge
}	
}	

- Change to regular merge mode.
- Signal merge_idx after all merge syntaxes.

There is no bitstream difference since VTM encoder doesn't select this condition.

Panasonic

Method2

- Coded as fallback merge mode when none of the merge modes are selected.

merge_data( ) {	
regular_merge_flag[x0][y0]	
if (regular_merge_flag[x0][y0] = = 1) {	
merge_idx[x0][y0]	Code regular merge
} else {	
mmvd_merge_flag[x0][y0]	
if (mmvd_merge_flag[x0][y0] = = 1) {	
・・・・	Code MMVD merge
} else {	
merge_subblock_flag[x0][y0]	
if (merge_subblock_flag[x0][y0] = = 1) {	
・・・・	Code sub block merge
} else {	
ciip_flag[x0][y0]	
if (ciip_flag[x0][y0] = = 1) {	
・・・・	Code CIIP merge
} else if (triangle conditions are true){	
・・・・	Code triangle merge
} else {	
FallBackMergeFlag[x0][y0] = 1	Code fallback merge
}	
}	
}	
}	
}	

- Coded as fallback merge mode.

  inter_pred_idc = PRED_L0

  ref_idx_l0 = 0

  MV = (0,0)

There is no bitstream difference since VTM encoder doesn't select this condition.

Panasonic

Method3

Revert to VTM4.0 syntax

merge_data( ) {	
mmvd_merge_flag[x0][y0]	
if (mmvd_merge_flag[x0][y0] = = 1) {	
. . . .	Code MMVD merge
} else {	
merge_subblock_flag[x0][y0]	
if (merge_subblock_flag[x0][y0] = = 1) {	
. . . .	Code sub block merge
} else {	
ciip_flag[x0][y0]	
if (ciip_flag[x0][y0] = = 1) {	
. . . .	Code CIIP merge
} else {	
merge_triangle_flag[x0][y0]	
if (merge_triangle_flag[x0][y0] = = 1) {	
. . . .	Code triangle merge
} else {	
merge_idx[x0][y0]	Code regular merge
}	
}	
}	
}	
}	

- Remove regular_merge_flag and revert to merge_triangle_flag.
- Regular merge mode is detected when merge_triangle_flag is equal to 0.

Panasonic

Simulation results

- Method1 and Method2

  Both results are not changed from VTM5.0.

- Method3

  Some loss is observed.

	RA (Over VTM-5.0)				
	Y	U	V	EncT	DecT
Class A1	-0.06%	-0.29%	-0.07%	101%	100%
Class A2	-0.02%	-0.02%	-0.02%	100%	100%
Class B	0.05%	0.12%	-0.15%	101%	100%
Class C	0.09%	0.01%	-0.06%	100%	100%
Class E					
Overall	0.03%	-0.02%	-0.08%	101%	100%
Class D	0.20%	0.27%	0.52%	101%	97%
Class F	0.15%	0.20%	0.15%	100%	101%

	LDB (Over VTM-5.0)				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	0.05%	0.17%	0.22%	100%	101%
Class C	0.05%	0.03%	0.39%	101%	104%
Class E	0.40%	0.86%	0.39%	100%	116%
Overall	0.14%	0.30%	0.32%	100%	106%
Class D	-0.15%	-0.16%	1.07%	101%	104%
Class F	0.66%	0.97%	0.22%	100%	111%

Conclusion

- 3 methods are proposed to solve undefined condition of merge mode syntax.
  - (1) Coded as regular merge mode in the condition.
  - (2) Coded as fallback merge mode in the condition.
  - (3) Revert to VTM4.0 syntax.

- Method (1) has the least amount of change for the existing specification and no bitstream difference.

We propose to adopt method (1) into VVC.